

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

Zemnieku tirgus sistēma

Kvalifikācijas darbs

Autors: Kristaps Bērziņš kb13054

Vadītājs: doc. Edgars Rencis

Rīga 2015

ANOTĀCIJA

Šajā dokumentā ir iekļauta kvalifikācijas darba ietvaros izstrādātas, tīmekļa bāzētas, Zemnieku tirgus sistēmas tehniskā dokumentācija, kas ietver prasību specifikāciju, projektējuma aprakstu, testēšanas dokumentāciju un pirmkoda fragmentus. Papildus dokumentā var iepazīties ar sistēmas projekta organizāciju, konfigurāciju pārvaldību un darbietilpības novērtējumu.

Zemnieku tirgus sistēmas ir tiešsaistes lietotne, kuras mērķis ir uzlabot saikni starp lauksaimniekiem un patērētājiem. Lietotne kalpo kā reģistrēto lauksaimnieku datu bāze, kur interesentiem ir iespēja iepazīties ar lauksaimnieku piedāvātajiem zemnieku saimniecībā audzētajiem produktiem.

Atslēgas vārdi:

Laravel 5, tīmekļa vietne, zemnieku tirgus.

ABSTRACT

This qualification work includes web based, Farmers market system technical documentation, which shall include a description of the requirements of the specification, design, testing, documentation, and source code fragments. In addition to the document consists of systems, configuration management and organisation of the assessment of efficiency.

Farmers market system is an online web site, system aims to improve the links between farmers and consumers. The application serves as the registered farmers database where interested parties have the opportunity to familiarise themselves with the farmers and farm products offers.

Key words:

Laravel 5, web site, farmers market.

SATURA RĀDĪTĀJS

ANOTĀCIJA.....	2
ABSTRACT.....	3
SATURA RĀDĪTĀJS	4
VĀRDNĪCA.....	9
IEVADS	10
1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	10
1.1. Ievads.....	10
1.1.1. Dokumenta nolūks	10
1.1.2. Darbības sfēra	10
1.1.3. Saistība ar citiem dokumentiem.....	10
1.1.4. Dokumenta pārskats	10
1.2. Vispārējs apraksts.....	11
1.2.1. Produkta funkcijas.....	11
1.2.2. Produkta perspektīva.....	11
1.2.3. Lietotāja raksturiezīmes	11
1.2.3.1. Neregistrēts lietotājs	11
1.2.3.2. Dalībnieks.....	12
1.2.3.3. Sistēmas pārvaldnieks.....	12
1.2.4. Vispārīgie ierobežojumi.....	12
1.2.5. Pieņemumi un atkarības	13
1.3. Funkcionālās prasības.....	13
1.3.1. Dalībnieka reģistrēšanās	13
1.3.2. Dalībnieka iežurnālēšanās darbam sistēmā	14
1.3.3. Dalībnieka izžurnālēšanās no darba sistēmā	15
1.3.4. Dalībnieka profila rediģēšana.....	15
1.3.5. Dalībnieka paroles maiņa	16
1.3.6. Dalībnieka nozaudētās paroles atjaunošana.....	17
1.3.7. Grupas izveidošana	17
1.3.8. Grupas satura rediģēšana	18
1.3.9. Grupas dzēšana	19
1.3.10. Zemnieku lietotāja profila rediģēšana.....	20
1.3.11. Zemnieku lietotāja profila dzēšana.....	21
1.3.12. Zemnieka produkta pievienošana	21
1.3.13. Zemnieka produkta rediģēšana	22

1.3.14.	<i>Zemnieka produkta dzēšana</i>	23
1.3.15.	<i>Grupas dalībnieka produktu pievienošana produktu grozam</i>	24
1.3.16.	<i>Grupas dalībnieka produktu dzēšana no produktu groza</i>	24
1.3.17.	<i>Grupas dalībnieka individuālā pasūtījuma apstiprināšana</i>	25
1.3.18.	<i>Grupas dalībnieka individuālā pasūtījuma dzēšana</i>	26
1.3.19.	<i>Grupas pasūtījuma apstiprināšana</i>	26
1.3.20.	<i>Grupas pasūtījuma noraidīšana</i>	27
1.4.	<i>Ārējā saskarne</i>	28
1.4.1.	<i>Lietotāju saskarne</i>	28
1.4.2.	<i>Programmatūras saskarne</i>	28
1.5.	<i>Nefunkcionālās prasības</i>	28
1.5.1.	<i>Veiktspējas prasības</i>	28
1.5.2.	<i>Drošības prasības</i>	28
1.5.3.	<i>Sistēmas pieejamības prasības</i>	29
1.5.4.	<i>Aizsardzība pret nesankcionētu piekļuvi</i>	29
2.	PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS	29
2.1.	<i>Ievads</i>	29
2.1.1.	<i>Nolūks</i>	29
2.1.2.	<i>Darbības sfēra</i>	29
2.1.3.	<i>Saistība ar citiem dokumentiem</i>	29
2.2.	<i>Datu bāzes tabulu detalizēts apraksts</i>	29
2.2.1.	<i>Tabula 'users'</i>	29
2.2.2.	<i>Tabula 'password_resets'</i>	30
2.2.3.	<i>Tabula 'user_types'</i>	30
2.2.4.	<i>Tabula 'groups'</i>	30
2.2.5.	<i>Tabula 'orders'</i>	31
2.2.6.	<i>Starptabula 'order_items'</i>	31
2.2.7.	<i>Tabula 'products'</i>	31
2.2.8.	<i>Tabula 'categories'</i>	32
2.3.	<i>Dekompozīcijas apraksts</i>	32
2.3.1.	<i>Moduļu dekompozīcija</i>	32
2.4.	<i>Atkarību apraksts</i>	33
2.4.1.	<i>Datu atkarības – Fiziskais ER modelis</i>	33
2.5.	<i>Saskarnes apraksts</i>	34
2.5.1.	<i>Kontrolieru funkciju saskarne</i>	34
2.5.1.1.	<i>Auth/AuthController</i>	34

2.5.1.2.	Auth/PasswordController	34
2.5.1.3.	HomeController	34
2.5.1.4.	WelcomeController	35
2.5.1.5.	UserController	35
2.5.1.6.	GroupController	36
2.5.1.7.	ProductController	38
2.5.1.8.	CategoryController	40
2.5.1.9.	OrderController	41
2.5.2.	<i>Skati</i>	41
3.	TESTĒŠANAS DOKUMENTĀCIJA	42
3.1.	Testēšanas procesa apraksts	42
3.2.	Vienībtestēšanas rezultāti	43
3.2.1.	<i>Kontrolieris WelcomeController</i>	43
3.2.1.1.	__construct	43
3.2.1.2.	index	43
3.2.2.	<i>Kontrolieris HomeController</i>	43
3.2.2.1.	__construct	43
3.2.2.2.	index	43
3.2.2.3.	farmers	43
3.2.3.	<i>Kontrolieris UserController</i>	44
3.2.3.1.	__construct	44
3.2.3.2.	show	44
3.2.3.3.	edit	44
3.2.3.4.	update	45
3.2.3.5.	destroy	46
3.2.4.	<i>Kontrolieris GroupController</i>	46
3.2.4.1.	__construct	46
3.2.4.2.	index	46
3.2.4.3.	create	46
3.2.4.4.	store	47
3.2.4.5.	show	47
3.2.4.6.	edit	48
3.2.4.7.	update	48
3.2.4.8.	join	49
3.2.4.9.	leave	49
3.2.4.10.	orders	49

3.2.4.11.	postOrders.....	50
3.2.4.12.	destroy	50
3.2.5.	<i>Kontrolieris ProductController</i>	50
3.2.5.1.	__construct	50
3.2.5.2.	index	50
3.2.5.3.	create.....	50
3.2.5.4.	store	51
3.2.5.5.	edit	51
3.2.5.6.	show.....	52
3.2.5.7.	update	52
3.2.5.8.	destroy	53
3.2.6.	<i>Kontrolieris CategoryController</i>	53
3.2.6.1.	__construct	53
3.2.6.2.	index	53
3.2.6.3.	create.....	53
3.2.6.4.	store	54
3.2.6.5.	edit	54
3.2.6.6.	update	54
3.2.6.7.	destroy	55
3.2.7.	<i>Kontrolieris OrderController</i>	55
3.2.7.1.	__construct	55
3.2.7.2.	getCart	55
3.2.7.3.	getCartadd.....	56
3.2.7.4.	getCartremove	56
3.2.7.5.	getPlace.....	56
3.2.7.6.	postPlace.....	56
3.2.7.7.	destroy	56
4.	PROJEKTA ORGANIZĀCIJA.....	57
5.	KONFIGURĀCIJAS PĀRVALDĪBA	58
6.	DARBIETILPĪBAS NOVĒRTĒJUMS.....	59
6.1.	Sākotnējais darbietilpības novērtējums	59
6.2.	Darbietilpības novērtējums pēc sistēmas realizācijas.....	59
7.	KVALITĀTES NODROŠINĀŠANA	62
	NOBEIGUMS	63
	IZMANTOTĀ LITERATŪRA	64
	PIELIKUMI.....	65

Zemnieku tirgus sistēmas pirmkoda fragmenti	65
Kontrolieris.....	65
Modelis.....	69
Skats	70

VĀRDNĪCA

PPA – Programmatūras projektējuma apraksts. Dokumentēts *Zemnieku tirgus sistēmas* tehniskais projektējums.

PPS – Programmatūras prasību specifikācija. Dokumentēta *Zemnieku tirgus sistēmas* funkcionalitāte un citas prasības.

Laravel – Valodas PHP izstrādes satvars.

MySQL – atvērtā pirmkoda datu bāzes vadības sistēma.

SQL – datu bāzu vaicājumu valoda.

UTF-8 – teksta kodēšanas metode, ko izmanto rakstzīmju un citu simbolu attēlošanai unikoda standartā.

MVC – Model View Controller arhitektūra ir veids, kā strukturizēt projekta kodu, lai sadalītu to trīs pamata komponentēs.

DPD – Datu Plūsmu Diagramma. Attēlo sistēmā notiekošās darbības, vai komponentes, kuras veic darbības, un datu plūsmu starp tām, kā arī attēlo lietotāju un ārējo ierīču sadarbību ar sistēmu un otrādi.

LVS - Latvijas Valsts Standarts.

IEVADS

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1. Dokumenta nolūks

Programmatūras prasību specifikācijas (PPS) nolūks ir precīzi un viennozīmīgi aprakstīt visas ‘Zemnieku tirgus sistēmas’ prasības un mērķus, pielietojumu, kā arī funkcionālās un nefunkcionālās prasības.

PPS ir paredzēta gan programmatūras pasūtītājiem, gan tās izstrādātājiem. Pēc šī dokumenta vadlīnijām tiks izstrādāta *Zemnieku tirgus sistēma*.

1.1.2. Darbības sfēra

Zemnieku tirgus sistēmas ir tiešsaistes lietotne, kuras mērķis ir uzlabot saikni starp lauksaimniekiem un patērētājiem. Lietotne kalpo kā reģistrēto lauksaimnieku datu bāze, kur interesentiem ir iespēja iepazīties ar lauksaimnieku piedāvātajiem zemnieku saimniecībā audzētajiem produktiem.

1.1.3. Saistība ar citiem dokumentiem

Dokuments sastādīts vadoties pēc standarta LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis”[7] un LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai” [6] prasības.

1.1.4. Dokumenta pārskats

Programmatūra prasību specifikācija sastāv no piecām nodaļām:

- Ievads. Satur informāciju par dokumenta nolūku, darbības sfēru un saistību ar citiem dokumentiem.
- Vispārējs apraksts. Sniedz vispārīgu informāciju par produkta funkcionalitāti, lietotāja raksturojumiem, kā arī sistēmas vispārīgajiem ierobežojumiem.

- Funkcionālās prasības. Sniedz detalizētu informāciju par funkcionalitāti sistēmā.
- Ārējā saskarne. Sniedz informāciju par sistēmas lietotāju saskarnes principiem.
- Nefunkcionālās prasības. Sniedz informāciju par prasībām, kas attiecināmas uz sistēmas darbību kā, piemēram, veiktspējas un drošības prasības.

1.2. Vispārējs apraksts

1.2.1. Produkta funkcijas

Izstrādātajā sistēmā tiek nodrošināta sekojoša funkcionalitāte:

- Lietotāju funkcijas
 - Lietotāja reģistrācija un autentifikācija
 - Lietotāja atteikšanās
 - Lietotāja profils
 - Lietotāja datu labošana
 - Lietotāja produktu groza izveidošana, rediģēšana, dzēšana
- Grupu funkcijas
 - Grupas pievienošana, rediģēšana, dzēšana
 - Lietotāju pievienošana, noraidīšana grupai
 - Grupas pasūtījuma noformēšana
- Zemnieku funkcijas
 - Zemnieka lietotāja profila labošana, dzēšana
 - Produktu pievienošana, rediģēšana, dzēšana

1.2.2. Produkta perspektīva

Programmatūra būs tipiska tīmekļa tehnoloģiju balstīta sistēma. Sistēma tiks novietota uz mākoņa servera, tādejādi sistēma būs pieejama lietotājiem jebkurā laikā. Sistēma būs kā starpnieks starp patērētājiem un gala produkta piedāvātājiem, tādēļ sistēmu izmantos abas iepriekšminētās puses.

1.2.3. Lietotāja raksturiezīmes

1.2.3.1. Neregistrēts lietotājs

Šīs grupas lietotājiem ir nepieciešamas prasmes strādāt ar datoru, nokonfigurēts interneta pieslēgums, moderns interneta pārlūks.

1.2.3.2. Dalībnieks

Papildus neregistrētā lietotāja minētajām prasībām, nepieciešams e-pasts un prasmes darbā ar to, lai izveidotu un aktivizētu lietotāja kontu.

1.2.3.3. Sistēmas pārvaldnieks

No sistēmas pārvaldnieka sagaidāma sistēmas procedūru pārzināšana un interneta drošības problēmu izpratne. Pārvaldnieka darba pienākumos ietilpst dažāda veida manipulācijas ar lietotāju personas datu apstrādi, tādejādi nepieciešams orientēties aktuālajos valsts likumos un spēt tos praksē ievērot. Pārvaldnieka darbs nosaka to, vai sistēma funkcionēs atbilstoši iecerētajām prasībām un spēs pildīt savu funkcionalitāti.

1.2.4. Vispārīgie ierobežojumi

- Jāņem vērā autortiesību likumdošana valstī, kurā šī sistēma tiks fiziski ievietota.
- Sistēmas izstrādē drīkst izmantot tikai bezmaksas un atvērtā koda risinājumus
- Sistēmai jābūt bāzētai uz tīmekļa tehnoloģijām.
- Sistēmas darbība tiek nodrošināta, izmantojot tīmekļa pārlūkprogrammu, kas atbalsta HTML 5 un CSS 3 specificētos standartus, kā arī nodrošina JavaScript izpildi. Paredzēts, ka sistēma spēs pilnvērtīgi funkcionēt mūsdienīgās pārlūkprogrammās un operētājsistēmās, kas norādītas tabulā.

Darbstaciju operētājsistēmas	
Microsoft Windows 7,8	Apple OS X 10.9
<ul style="list-style-type: none">• Microsoft Internet Explorer 11• Mozilla Firefox 11.0• Google Chrome 9• Opera	<ul style="list-style-type: none">• Apple Safari 7• Mozilla FireFox 11.0• Google Chrome 9

- Normālai sistēmas funkcionēšanai, klienta izmantotās gala iekārtas minimālā ekrāna izšķirtspēja paredzama 1024 x 768 px darbstacijām.

1.2.5. Pieņēmumi un atkarības

- Sistēmas darbības videi tiek atvēlēts privātais serveris, kura konfigurāciju ir iespējams mainīt pēc sistēmas vajadzībām.
- Privātajam serverim tiks piešķirta statiska IP adrese, lai sistēma būtu pieejama gan iekšējā, gan ārējā tīklā.
- Sistēma neietekmē serverī paralēli strādājošo sistēmu darbību.
- Sistēmas serverim tiek nodrošināts, vismaz 100 Mbit/s ātrs interneta pieslēgums, lai lietotāju plūsma sistēmā neietekmētu sistēmas ātrdarbību

1.3. Funkcionālās prasības

1.3.1. Dalībnieka reģistrēšanās

Mērķis	
Funkcijas mērķis ir nodrošināt lietotājiem iespēju patstāvīgi veikt reģistrāciju, lai varētu izmantot sistēmas funkcijas, kuras to pieprasa. Funkcijas veic ievaddatu korektuma pārbaudi, pārlicinoties par e-pasta unikalitāti. Kļūdu gadījumā jā saglabā aizpildītie lauki. Paroles tīmeklī nevar tikt sūtītas brīvi lasāmā veidā.	
Identifikators	
-	
Ievaddati	
Nosaukums	Piezīme
Vārds	Garumā no 3 līdz 30 simboliem
Uzvārds	Garumā no 3 līdz 30 simboliem
E-pasts	Atbilst RFC 5322 standartam [8]
Parole	Garumā vismaz 3 jebkādi simboli
Atkārtota parole	Garumā vismaz 3 jebkādi simboli
Apstrāde	
Dalībnieka reģistrācijas procesā veic sekojošas pārbaudes un darbības: <ul style="list-style-type: none">• Vai vārds atbilst ievaddatu prasībām;• Vai uzvārds atbilst ievaddatu prasībām;• Vai e-pasts atbilst ievaddatu prasībām;• Vai paroles sakrīt un atbilst ievaddatu prasībām;• Vai dalībnieks ar šādu e-pastu jau nav reģistrēts;• Datubāzē tiek pievienots jauns lietotāja ieraksts;	

Izvaddati
Pēc sekmīgas reģistrēšanās tiek atvērta dalībnieka profils un parādīts paziņojums: “Reģistrācija notikusi veiksmīgi!”
Kļūdas paziņojumi
Reģistrēšanās procesā var rasties sekojoši kļūdas paziņojumi: <ul style="list-style-type: none"> • “Vārds var sastāvēt tikai no Latīņu burtiem un simboliem”; • “Uzvārds var sastāvēt tikai no Latīņu burtiem un simboliem”; • “Paroles nesakrīt!”; • “Parolei jābūt vismaz 3 simbolus garai!”; • “Nederīga e-pasta adrese!”; • “Šis e-pasts jau ir reģistrēts sistēmā!”;

1.3.2. Dalībnieka iežurnālēšanās darbam sistēmā

Mērķis	
Funkcijas mērķis ir ļaut reģistrētam un aktivizētam dalībniekam pieteikties darbam sistēmā, izmantojot savu lietotājvārdu un paroli.	
Identifikators	
-	
Ievaddati	
Nosaukums	Piezīme
E-pasts	Iepriekš definēta vērtība
Parole	Iepriekš definēta vērtība
Apstrāde	
Sistēma veic sekojošās pārbaudes un darbības: <ul style="list-style-type: none"> • Vai ir norādīts e-pasts; • Vai datubāzē eksistē dalībnieks ar šādu e-pastu; • Vai parole atbilst e-pastam; • Saglabā datubāzē jaunu dalībnieka sesiju. 	
Izvaddati	
Dalībnieks tiek novirzīts uz vietnes sākuma lapu.	
Kļūdas paziņojumi	
Pieteikšanās procesā var rasties sekojoši kļūdas paziņojumi:	

<ul style="list-style-type: none"> • “Lūdzu, ievadiet e-pastu!”; • “Lietotājs ar šādu e-pastu nav atrasts!” • “Lūdzu, ievadiet paroli!”; • “E-pasts un/vai parole nav derīga!”.

1.3.3. Dalībnieka izžurnālēšanās no darba sistēmā

Mērķis	
Funkcijas mērķis ir ļaut dalībniekam pārtraukt sesiju sistēmā un dzēst sesijas datus.	
Identifikators	
-	
Ievaddati	
Nosaukums	Piezīme
Izžurnālēšanās pieprasījuma saite	
Apstrāde	
Sistēma dzēš dalībnieka sesijas datus.	
Izvaddati	
Lietotājs tiek novirzīts uz vietnes sākuma lapu un tiek attēlots paziņojums “Darbs sistēmā veiksmīgi pabeigts!”.	
Kļūdas paziņojumi	
-	

1.3.4. Dalībnieka profila rediģēšana

Mērķis	
Funkcijas mērķis ir ļaut dalībniekam rediģēt savu profilu.	
Identifikators	
UserController@edit	
Ievaddati	
Nosaukums	Piezīme
Uzvārds	Garumā no 3 līdz 30 simboliem
E-pasts	Atbilst RFC 5322 standartam
Kontakttālrunis	Ciparu virkne
Apstrāde	

Sistēma veic sekojošās pārbaudes un darbības: <ul style="list-style-type: none"> • Pārbauda vai dalībnieks rediģē savu profilu. • Pārbauda vai neeksistē jau tāds epasts. • Rediģē ierakstu datubāzē
Izvaddati
Dalībnieks tiek novirzīts uz sava profila lapu ar paziņojumu: “Profils veiksmīgi labots!”.
Kļūdas paziņojumi
Dalībnieka profila rediģēšanas procesā var rasties sekojoši kļūdas paziņojumi: <ul style="list-style-type: none"> • “Lietotājs ar šādu nosaukumu jau eksistē!”; • “Lauks ‘Uzvārds’ ir obligāts!”; • “Lauks ‘E-pasts’ ir obligāts!”; • “Lauks ‘Lokācija’ ir obligāts!”;

1.3.5. Dalībnieka paroles maiņa

Mērķis	
Funkcijas mērķis ir ļaut lietotājam patstāvīgi nomainīt lietotāja paroli darbam ar sistēmu.	
Identifikators	
UserController@getPasswordchange	
Ievaddati	
Nosaukums	Piezīme
Vecā parole	Iepriekš definēta vērtība
Jaunā parole	Garumā vismaz 3 jebkādi simboli
Jaunā parole atkārtoti	Sakrīt ar iepriekš ievadīto paroli.
Apstrāde	
<ul style="list-style-type: none"> • Pārbauda vai vecā parole sakrīt ar esošo paroli datubāzē; • Pārbauda vai ievadītās jaunās paroles sakrīt; • Ieraksta jauno paroli datubāzē 	
Izvaddati	
Dalībnieks tiek novirzīts uz sava profila lapu ar paziņojumu: “Parole veiksmīgi nomainīta!”	

Kļūdas paziņojumi
Dalībnieka paroles maiņas procesā var rasties sekojoši kļūdas paziņojumi: <ul style="list-style-type: none"> • “Ievadītā parole nav pareiza!” • “Paroles nesakrīt!”

1.3.6. Dalībnieka nozaudētās paroles atjaunošana

Mērķis	
Funkcijas mērķis ir ļaut sistēmas dalībniekiem atjaunot paroli, saņemot uz e-pastu reģistrācijas procesā izmantoto paroles atjaunošanas saiti.	
Identifikators	
Password_reset	
Ievaddati	
Nosaukums	Piezīme
E-pasts	Iepriekš definēta vērtība
Apstrāde	
Paroles atjaunošanas procesā jāveic šādas darbības un pārbaudes: <ul style="list-style-type: none"> • Vai sistēmā eksistē lietotāja ieraksts ar šādu e-pastu; • Uz dalībnieka e-pastu tiek sūtīta unikāla saite paroles atjaunošanai; • Sekojot šai saitei, dalībnieka sesijas tiek iznīcinātas. 	
Izvaddati	
Pēc pieprasījuma formas nosūtīšanas tiek attēlots paziņojums: “Saite paroles atjaunošanai nosūtīta uz e-pastu!”.	
Pēc saites apmeklēšanas dalībnieks tiek novirzīts uz paroles maiņas formu.	
Kļūdas paziņojumi	
Dalībnieka nozaudētas paroles atjaunošanas procesā var rasties sekojoši kļūdas paziņojumi: <ul style="list-style-type: none"> • “Lietotājs ar šādu e-pastu nav atrasts!”; 	

1.3.7. Grupas izveidošana

Mērķis
Funkcija mērķis ir ļaut sistēmas dalībniekam izveidot grupu, kurai varēs pievienoties citi sistēmas dalībnieki un veidot kopīgus pasūtījumus.
Identifikators

GroupController@create	
Ievaddati	
Nosaukums	Piezīme
Nosaukums	Garumā no 3 līdz 30 simboliem
Apraksts	Garumā no 3 līdz 254 simboliem
Lokācija	Garumā no 3 līdz 254 simboliem
Apstrāde	
<ul style="list-style-type: none"> • Vai lietotājs ir pieteicies darbam ar sistēmu; • Vai grupa ar tādu nosaukumu jau neeksistē; • Vai visi lauki ir aizpildīti; • Datubāzēs tabulā 'groups' izveido jaunu ierakstu; 	
Izvaddati	
Veiksmīgas apstrādes rezultātā sistēmas dalībnieks tiek pāradresēts uz izveidotās grupas sākuma lapu ar paziņojumu 'Grupa veiksmīgi izveidota!'	
Kļūdas paziņojumi	
<p>Grupas veidošanas procesā var rasties sekojoši kļūdas paziņojumi:</p> <ul style="list-style-type: none"> • "Grupa ar šādu nosaukumu jau eksistē!"; • "Lauks 'Nosaukums' ir obligāts!"; • "Lauks 'Apraksts' ir obligāts!"; • "Lauks 'Lokācija' ir obligāts!"; 	

1.3.8. Grupas saturs rediģēšana

Mērķis	
Funkcijas mērķis ir, izmaiņu gadījumā, kas saistītas ar grupu, ļaut dalībniekam rediģēt grupas informatīvo saturu.	
Identifikators	
GroupController@edit	
Ievaddati	
Nosaukums	Piezīme
Nosaukums	Garumā no 3 līdz 30 simboliem
Apraksts	Garumā no 3 līdz 254 simboliem
Lokācija	Garumā no 3 līdz 254 simboliem

Apstrāde
<ul style="list-style-type: none"> • Vai lietotājs ir grupas īpašnieks; • Vai grupa ar tādu nosaukumu jau eksistē; • Vai visi lauki ir aizpildīti; • Datubāzes tabulā ‘groups’ rediģē ierakstu ar attiecīgo identifikatoru;
Izvaddati
Veiksmīgas apstrādes rezultātā sistēmas dalībnieks tiek pāradresēts uz grupas sākuma lapu ar paziņojumu ‘Grupa veiksmīgi atjaunota!’
Kļūdas paziņojumi
<p>Grupas rediģēšanas procesā var rasties sekojoši kļūdas paziņojumi:</p> <ul style="list-style-type: none"> • “Grupa ar šādu nosaukumu jau eksistē!”; • “Lauks ‘Nosaukums’ ir obligāts!”; • “Lauks ‘Apraksts’ ir obligāts!”; • “Lauks ‘Lokācija’ ir obligāts!”;

1.3.9. Grupas dzēšana

Mērķis	
Funkcijas mērķis ir grupas neaktivitātes, domstarpību vai jebkādu citu iemeslu dēļ grupu dzēst no sistēmas un sniegt iespēju lietotājiem meklēt citu grupu, kurai pievienoties.	
Identifikators	
GroupController@destroy	
Ievaddati	
Nosaukums	Piezīme
Grupas dzēšanas pieprasījums saites veidā.	
Apstrāde	
<ul style="list-style-type: none"> • Sistēma dzēš grupas dalībnieku saites ar grupu; • Sistēma dzēš īpašnieka saiti ar grupu; • Sistēma dzēš ierakstu no datubāzes tabulas ‘groups’ ar attiecīgo identifikatoru; 	
Izvaddati	

Veiksmīgas grupas dzēšanas rezultātā sistēmas dalībnieks tiek pāradresēts uz sistēmas sākuma lapu ar paziņojumu ‘Grupa veiksmīgi izdzēsta!’
Kļūdas paziņojumi
<p>Grupas dzēšanas procesā var rasties sekojoši kļūdas paziņojumi:</p> <ul style="list-style-type: none"> • “Neizdevās dzēst grupu”;

1.3.10. Zemnieku lietotāja profila rediģēšana

Mērķis	
Funkcijas mērķis ir sniegt iespēju rediģēt zemnieka profila pamatinformāciju.	
Identifikators	
UserController@edit	
Ievaddati	
Nosaukums	Piezīme
Nosaukums	Garumā no 3 līdz 30 simboliem
E-pasts	Atbilst RFC 5322 standartam
Kontaktārunis	Ciparu virkne
Apraksts	Garumā no 3 līdz 254 simboliem
Lokācija	Garumā no 3 līdz 254 simboliem
Apstrāde	
<ul style="list-style-type: none"> • Pārbauda vai sistēmas dalībnieks ir zemnieks • Pārbauda vai zemnieks rediģē savu profilu. • Pārbauda vai neeksistē jau tāds nosaukums. • Pārbauda vai neeksistē jau tāds e-pasts • Rediģē ierakstu datubāzē 	
Izvaddati	
Veiksmīgas apstrādes rezultātā dalībnieks tiek pāradresēts uz sava profila sākuma lapu ar paziņojumu ‘Profils veiksmīgi atjaunots!’.	
Kļūdas paziņojumi	
<p>Zemnieka profila rediģēšanas procesā var rasties sekojoši kļūdas paziņojumi:</p> <ul style="list-style-type: none"> • “Šāds nosaukums jau eksistē!”; • “Lietotājs ar šādu E-pastu jau eksistē.” • “Lauks ‘Nosaukums’ ir obligāts!”; 	

- “Lauks ‘Apraksts’ ir obligāts!”;
- “Lauks ‘Lokācija’ ir obligāts!”;

1.3.11. Zemnieku lietotāja profila dzēšana

Mērķis	
Funkcijas mērķis ir ļaut dalībniekam izdzēst savu zemnieka profilu.	
Identifikators	
UserController@destroy	
Ievaddati	
Nosaukums	Piezīme
Profila dzēšanas pieprasījums saites veidā.	
Apstrāde	
<ul style="list-style-type: none"> • Pārbauda vai dalībniekam ir tiesības veikt šādu darbību; • Dzēš visas dalībnieka saites ar izveidotajiem produktiem; • Dzēš dalībnieka profilu; 	
Izvaddati	
Veiksmīgas apstrādes rezultātā dalībnieks tiek izžurnalēts no sistēmas un pārdresēts uz sākuma lapu ar paziņojumu ‘Profils veiksmīgi dzēsts!’;	
Kļūdas paziņojumi	

1.3.12. Zemnieka produkta pievienošana

Mērķis	
Funkcijas mērķis ir ļaut zemniekiem pievienot savas zemnieku saimniecības piedāvāto produkciju.	
Identifikators	
ProductController@create	
Ievaddati	
Nosaukums	Piezīme
Nosaukums	Garumā no 3 līdz 30 simboliem
Apraksts	Garumā no 3 līdz 254 simboliem

Kategorija	Iepriekš definēta vērtība
Cena par kg	Decimāls skaitlis ar 2 cipariem aiz komata.
Bioloģisks	Iepriekš definēta vērtība
Apstrāde	
<ul style="list-style-type: none"> • Pārbauda vai sistēmas dalībnieks ir zemnieks; • Pārbauda vai visi lauki ir aizpildīti; • Pārbauda vai produkts ar tādu nosaukumu jau nav zemnieka piedāvājumā • Pievieno produktu datubāzē; 	
Izvaddati	
Veiksmīgas apstrādes rezultātā dalībnieks tiek pāradresēts uz sava profila sākuma lapu ar paziņojumu 'Produkts veiksmīgi pievienots'.	
Kļūdas paziņojumi	
Zemnieka produkta pievienošanas procesā var rasties sekojoši kļūdas paziņojumi: <ul style="list-style-type: none"> • "Šāds produkts jau pastāv!"; • "Lauks 'nosaukums' ir obligāts!"; • "Lauks 'apraksts' ir obligāts!"; • "Lauks 'kategorija' ir obligāts!"; • "Lauks 'cena' ir obligāts!" 	

1.3.13. Zemnieka produkta rediģēšana

Mērķis	
Funkcijas mērķis ir ļaut zemniekam rediģēt savus pievienotos produktus, piemēram, ja piedāvājuma apjoms ir mainījies, vai kādi no ievadlaukiem ievadīti nepareizi.	
Identifikators	
ProductController@edit	
Ievaddati	
Nosaukums	Piezīme
Nosaukums	Garumā no 3 līdz 30 simboliem
Apraksts	Garumā no 3 līdz 254 simboliem
Kategorija	Iepriekš definēta vērtība

Cena par kg	Decimāls skaitlis ar 2 cipariem aiz komata.
Bioloģisks	Iepriekš definēta vērtība
Apstrāde	
<ul style="list-style-type: none"> • Pārbauda vai sistēmas dalībnieks ir zemnieks; • Pārbauda vai visi lauki ir aizpildīti; • Pārbauda vai produkts ar tādu nosaukumu jau nav zemnieka piedāvājumā; • Rediģē ierakstu datubāzē; 	
Izvaddati	
Veiksmīgas apstrādes rezultātā dalībnieks tiek pāradresēts uz sava profila sākuma lapu ar paziņojumu ‘Produkts veiksmīgi atjaunots’.	
Kļūdas paziņojumi	
Zemnieka produkta rediģēšanas procesā var rasties sekojoši kļūdas paziņojumi: <ul style="list-style-type: none"> • “Šāds produkts jau pastāv!”; • “Lauks ‘nosaukums’ ir obligāts!”; • “Lauks ‘apraksts’ ir obligāts!”; • “Lauks ‘kategorija’ ir obligāts!”; • “Lauks ‘cena’ ir obligāts!” 	

1.3.14. Zemnieka produkta dzēšana

Mērķis	
Funkcijas mērķis ir ļaut zemniekam izņemt produktu no sava piedāvāto produktu klāsta.	
Identifikators	
ProductController@destroy	
Ievaddati	
Nosaukums	Piezīme
Produkta dzēšanas pieprasījums saites veidā.	
Apstrāde	
<ul style="list-style-type: none"> • Pārbauda vai dalībniekam ir tiesības veikt šādu darbību; • Dzēš starp tabulas ‘products’ ierakstu ar attiecīgo identifikācijas numuru; 	

Izvaddati
Veiksmīgas apstrādes rezultātā dalībnieks tiek pāradresēts uz sava profila sākuma lapu ar paziņojumu ‘Produkts veiksmīgi dzēsts’.
Kļūdas paziņojumi

1.3.15. Grupas dalībnieka produktu pievienošana produktu grozam

Mērķis	
Funkcijas mērķis ir sniegt iespēju dalībniekam, kurš atrodas kādā grupā veidot pasūtījumu pēc saviem ieskatiem.	
Identifikators	
OrderController@getCartadd	
Ievaddati	
Nosaukums	Piezīme
Nosaukums	Iepriekš definēta vērtība
Daudzums_kg	Decimāls skaitlis ar 2 cipariem aiz komata
Cena	Iepriekš definēta vērtība
Apstrāde	
<ul style="list-style-type: none"> • Pārbauda vai lietotājs ir grupā; • Pārbauda vai šis produkts no šī zemnieka jau nav iepirkuma grozā; 	
Izvaddati	
Veiksmīgas apstrādes rezultātā dalībnieks tiek pāradresēts uz savu produktu grozu ar paziņojumu ‘Produkts veiksmīgi pievienots grozam!’;	
Kļūdas paziņojumi	
Produkta pievienošanas grozam procesā var rasties sekojoši kļūdas paziņojumi: <ul style="list-style-type: none"> • “Neizdevās pievienot produktu!”; 	

1.3.16. Grupas dalībnieka produktu dzēšana no produktu groza

Mērķis
Funkcijas mērķis ir ļaut lietotājam dzēst produktu no produktu groza.
Identifikators

OrderController@getCartremove	
Ievaddati	
Nosaukums	Piezīme
Produkta dzēšanas pieprasījums no produktu groza saites veidā.	
Apstrāde	
<ul style="list-style-type: none"> Nosūta atkārtotu apstiprinājuma pieprasījumu par produkta dzēšanas lietotājam; Pēc apstiprinājuma saņemšanas izdzēš ierakstu no datubāzēs; 	
Izvaddati	
Veiksmīgas apstrādes rezultātā dalībnieks tiek pāradresēts uz savu produktu grozu ar paziņojumu ‘Produkts veiksmīgi dzēsts no groza!’;	
Kļūdas paziņojumi	
-	

1.3.17. Grupas dalībnieka individuālā pasūtījuma apstiprināšana

Mērķis	
Funkcijas mērķis ir ļaut dalībniekam, kas atrodas grupā apstiprināt savu pasūtījumu un pievienot to kopējam grupas pasūtījumam.	
Identifikators	
OrderController@getPlace	
Ievaddati	
Nosaukums	Piezīme
Produkta groza apstiprināšana saites veidā.	
Apstrāde	
<ul style="list-style-type: none"> Pārbauda vai lietotājs atrodas grupā; Pārbauda vai produktu grozā atrodas kāds produkts; Pievieno produktu grozu datubāzē; 	
Izvaddati	
Veiksmīgas apstrādes rezultātā dalībnieks tiek pāradresēts uz grupas sākuma lapu ar paziņojumu ‘Produktu grozs veiksmīgi noformēts!’	

Kļūdas paziņojumi
Produktu groza apstiprināšanas procesā var rasties sekojoši kļūdas paziņojumi <ul style="list-style-type: none"> • “Neizdevās! Produktu grozs ir tukšs.”

1.3.18. Grupas dalībnieka individuālā pasūtījuma dzēšana

Mērķis	
Funkcijas mērķis ir dot iespēju lietotājam uzreiz izdzēst visu pasūtījuma formu, nevis dzēst no tā pa vienam produktam.	
Identifikators	
OrderController@getRemove	
Ievaddati	
Nosaukums	Piezīme
Produkta groza dzēšanas apstiprinājums saites veidā.	
Apstrāde	
<ul style="list-style-type: none"> • Nosūta atkārtotu apstiprinājuma pieprasījumu par pasūtījuma dzēšanu lietotājam; • Pēc pieprasījuma saņemšanas dzēš saiti starp produktiem un pasūtījumu; • Dzēš lietotāja un pasūtījuma saiti; 	
Izvaddati	
Veiksmīgas apstrādes rezultātā dalībnieks tiek pāradresēts uz grupas sākuma lapu ar paziņojumu ‘Pasūtījums veiksmīgi dzēsts!’	
Kļūdas paziņojumi	
-	

1.3.19. Grupas pasūtījuma apstiprināšana

Mērķis	
Funkcijas mērķis ir apstiprināt grupas pasūtījumu un nosūtīt to zemniekam izpildei.	
Identifikators	
GroupController@placeOrder	
Ievaddati	
Nosaukums	Piezīme

Grupas pasūtījuma apstiprināšana saites veidā	
Apstrāde	
<ul style="list-style-type: none"> • Pārbauda vai apstiprinājumu veic grupas īpašnieks; • Pārbauda vai ir kāds pasūtījums; • Saglabā pasūtījumu datubāzē; • Nosūta zemniekiem informāciju par viņu esošajiem produktiem pasūtījumā; 	
Izvaddati	
Veiksmīgas apstrādes rezultātā dalībnieks tiek pāradresēts uz grupas sākuma lapu ar paziņojumu ‘Grupas pasūtījums veiksmīgi noformēts!’	
Kļūdas paziņojumi	
Grupas pasūtījuma apstiprināšanas procesā var rasties sekojoši kļūdas paziņojumi “Jums nav tiesību veikt šādu darbību”	

1.3.20. Grupas pasūtījuma noraidīšana

Mērķis	
Funkcijas mērķis ir dot iespēju grupas īpašniekam noraidīt grupas pasūtījumu.	
Identifikators	
Ievaddati	
Nosaukums	Piezīme
Grupas pasūtījuma noraidīšanas apstiprinājums saites veidā	
Apstrāde	
<ul style="list-style-type: none"> • Pārbauda vai apstiprinājumu veic grupas īpašnieks; • Pārbauda vai ir kāds pasūtījums; • Dzēš individuālos pasūtījumus no datubāzes; • Dzēš grupas pasūtījumu no datubāzes; • Nosūta ziņu grupas dalībniekiem; 	
Izvaddati	
Veiksmīgas apstrādes rezultātā dalībnieks tiek pāradresēts uz grupas sākuma lapu ar paziņojumu ‘Grupas pasūtījums veiksmīgi atcelts!’	

Kļūdas paziņojumi

Grupās pasūtījuma noraidīšanas procesā var rasties sekojoši kļūdas paziņojumi

- “Jums nav tiesību veikt šādu darbību”

1.4. Ārējā saskarne

1.4.1. Lietotāju saskarne

Lietotāju saskarne tiek realizēta, izmantojot tīmekļa pārlūkprogrammu. Ir jāparedz iespēja izvadīt sistēmas paziņojumus jebkurā vietnes lapā. Ārējā saskarne tiek veidota kā HTML lietojumprogramma.

Sistēmā ir nodrošinātas 4 lietotāju grupas:

- Administrators
- Reģistrēts lietotājs
- Zemnieks
- Viesis

Katrai lietotāju grupai ir pielāgota saskarne ar atbilstošām funkcijām.

1.4.2. Programmatūras saskarne

Sistēmas pieeja datu bāzei tiek nodrošināta ar MySQL servera palīdzību.

1.5. Nefunkcionālās prasības

1.5.1. Veiktspējas prasības

Sistēmai jānodrošina:

- Katras sistēmā paredzētās funkcijas izpildes laiku zem 5 sekundēm;
- Iespēju vienlaikus glabāt informāciju par vismaz 100 grupu ierakstiem datubāzē.
- Jāspēj nodrošināt vismaz 100 lietotāju vienlaicīgu uzturēšanos sistēmā.

Vienlaikus glabāt informāciju par vismaz 1000 pasūtījumu ierakstiem.

1.5.2. Drošības prasības

Sistēmai jānodrošina šifrēta parolu glabāšana datubāzē. Paroles nedrīkst tikt glabātas tekstuālā, atklātā veidā.

1.5.3. Sistēmas pieejamības prasības

Sistēmai jābūt pieejamai nepārtrauktā režīmā. Sistēmas uzlabojumus un atjaunošanas darbus, kad sistēma tiek ar nodomu īslaicīgi slēgta, jāparedz veikt dienas tumšajā daļā periodā no 22:00 līdz 6:00 pēc Austrumeiropas laika.

1.5.4. Aizsardzība pret nesankcionētu piekļuvi

Lietotāji sistēmā tiek autentificēti tikai ar reģistrācijas procesā uzrādīto e-pastu un paroli. Veiksmīga autorizācija var notikt tikai tad, ja tiek uzrādīta derīga e-pasta adrese un atbilstoša parole, pēc kuras lietotājs var uzsākt darbu sistēmā

2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1. Ievads

2.1.1. Nolūks

Programmatūras projektējuma apraksts (PPA) paredzēts izstrādājamās programmatūras “Zemnieku tirgus sistēma” projektējuma aprakstīšanai. Šis dokuments paredzēts sistēmas izstrādātājam.

2.1.2. Darbības sfēra

Šis dokuments apraksta “Zemnieku tirgus sistēmas” lietotāju sadalījumu tiesību grupās, sistēmas moduļus un to attiecības, datu dekompozīciju un datu atkarības.

2.1.3. Saistība ar citiem dokumentiem

Šis dokuments tika izstrādāts pēc LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma parakstīšanai” standarta. Dokuments tiek balstīts uz iepriekš izstrādātu PPS.

2.2. Datu bāzes tabulu detalizēts apraksts

2.2.1. Tabula ‘users’

Nosaukums	Datu tips	Atribūti	Oblig āts	Komentārs
id	INTEGER	PK – AI	Jā	Lietotāja identifikators
name	VARCHAR(255)		Jā	Lietotāja vārds

surname	VARCHAR(255)		Nē	Lietotāja uzvārds
telephone	INTEGER(11)		Nē	Lietotāja telefona numurs
email	VARCHAR(255)		Jā	Lietotāja e-pasts
password			Jā	Lietotāja parole
user_type_id	INTEGER(1)	FK ('user_types')	Jā	Lietotāja loma
group_id	INTEGER	FK ('groups')	Nē	Lietotāja grupas identifikators
title	VARCHAR(255)		Nē	Zemnieka saimniecības nosaukums
description	TEXT		Nē	Zemnieka saimniecības apraksts
location	VARCHAR(255)		Nē	Dalībnieka adrese

2.2.2. Tabula 'password_resets'

Nosaukums	Datu tips	Atribūti	Obligāts	Komentārs
email	VARCHAR(255)	PK	Jā	Dalībnieka e-pasts
token	VARCHAR(255)		Jā	

2.2.3. Tabula 'user_types'

Nosaukums	Datu tips	Atribūti	Obligāts	Komentārs
id	INTEGER	PK – AI	Jā	Lietotāja tipa identifikators
user_type	VARCHAR(255)		Jā	Lietotāja tipa nosaukums

2.2.4. Tabula 'groups'

Nosaukums	Datu tips	Atribūti	Obligāts	Komentārs
id	INTEGER	PK – AI	Jā	Grupas identifikators

title	VARCHAR(255)		Jā	Grupas nosaukums
description	TEXT		Jā	Grupas apraksts
location	VARCHAR(255)		Jā	Grupas adrese
owner_id	INTEGER	FK('users')	Jā	Grupas izveidotāja identifikators

2.2.5. Tabula 'orders'

Nosaukums	Datu tips	Atribūti	Obligāts	Komentārs
id	INTEGER	PK – AI	Jā	Pasūtījuma identifikators
user_id	INTEGER	FK('users')	Jā	Pasūtījuma lietotāja identifikators
group_id	INTEGER	FK('groups')	Jā	Pasūtījuma grupas identifikators
status	BOOLEAN	Default('false')	Jā	Pasūtījuma izpildes status.

2.2.6. Starptabula 'order_items'

Nosaukums	Datu tips	Atribūti	Obligāts	Komentārs
id	INTEGER	PK – AI	Jā	Pasūtījuma produkta identifikators
order_id	INTEGER	FK('orders')	Jā	Pasūtījuma identifikators
product_id	INTEGER	FK('products')	Jā	Produkta identifikators
quantity_kg	Float(5,2)		Jā	Produkta daudzums kilogramos
price	Float(5,2)		Jā	Produkta cena gabalos vai kilogramos

2.2.7. Tabula 'products'

Nosaukums	Datu tips	Atribūti	Obligāts	Komentārs
-----------	-----------	----------	----------	-----------

id	INTEGER	PK – AI	Jā	Produkta identifikators
name	VARCHAR(255)		Jā	Produkta nosaukums
description	TEXT		Jā	Produkta apraksts
bio	boolean	Default('false')	Nē	Produkta paternitāte (bioloģisks)
category_id	INTEGER	FK('categories')	Jā	Produkta kategorija
price	Float(5,2)		Jā	Produkta cena
user_id	INTEGER	FK('users')	Jā	Produkta zemnieka identifikators

2.2.8. Tabula 'categories'

Nosaukums	Datu tips	Atribūti	Obligāts	Komentārs
id	INTEGER	PK – AI	Jā	Kategorijas identifikators
name	VARCHAR(255)		Jā	Kategorijas nosaukums

2.3. Dekompozīcijas apraksts

2.3.1. Moduļu dekompozīcija

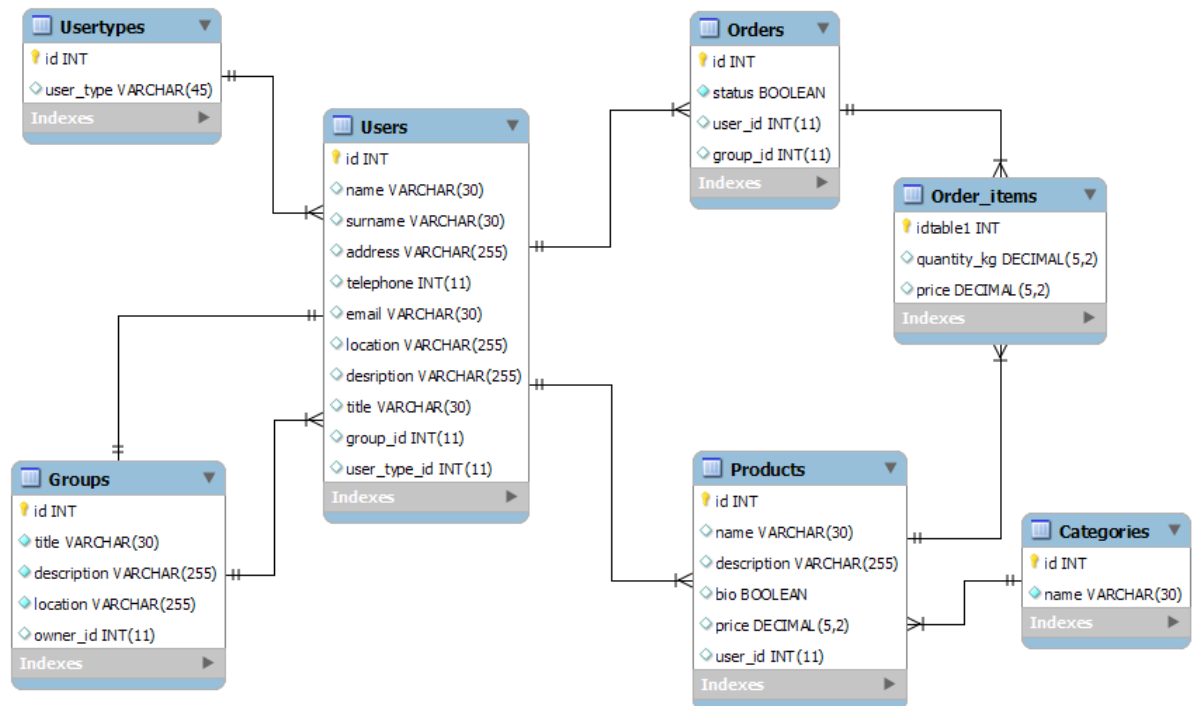
Šajā sadaļā tiek aprakstīta PPS aprakstīto funkcionālo prasību dekompozīcija moduļos.

Aprakstīto modeļu mērķis ir nodrošināt sistēmas kontrolieru klašu metožu darbu ar datu bāzi. Katrai datu bāzes tabulai atbilst viens modelis. Kolonnas '*belongs_to*' un '*has_many*' atsevišķi parāda datu atkarības starp tabulām, kas tiek attēlotas satvarā Laravel un tiek definētas katrā modelī.

Modeļa nosaukums	Tabula	Belongs_to	Has_many
App\User	'create_users_table'	'App\Group' 'App\UserType'	'App\Product' 'App\Order'
App\UserType	'create_user_types_table'		'App\User'
App\Group	'create_groups_table'	'App\User'	'App\User' 'App\Order'
App\Category	'create_categories_table'		'App\Product'
App\Product	'create_products_table'	'App\Category' 'App\User'	'App\OrderItems'
App\Order	'create_orders_table'	'App\User' 'App\Group'	'App\OrderItem'
App\OrderItem	'create_order_items_table'	'App\Order' 'App\Product'	

2.4. Atkarību apraksts

2.4.1. Datu atkarības – Fiziskais ER modelis



2.5. Saskarnes apraksts

2.5.1. Kontrolieru funkciju saskarne

2.5.1.1. Auth/AuthController

AuthController ir Laravel 5 iebūvētais kontrolieris, kas nodrošina lietotāja reģistrāciju. No darba autora puses nekādas izmaiņas šim kontrolierim netika veiktas.

Metodes nosaukums	GET parametri	Apraksts
__construct		Izsauc metodi viesu reģistrācijai sistēmā.

2.5.1.2. Auth/PasswordController

PasswordController ir Laravel 5 iebūvētais kontrolieris, kas nodrošina lietotāja paroles atjaunošanu un autentifikāciju un ar to saistītos sistēmas drošības pasākumus. No darba autora puses nekādas izmaiņas šim kontrolierim netika veiktas.

Metodes nosaukums	GET parametri	Apraksts
__construct		Izsauc metodi viesu autentifikācijai sistēmā.

2.5.1.3. HomeController

Metodes nosaukums	GET parametri	Apraksts
__construct		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas. Šim kontrolierim – tikai reģistrēti lietotāji.
index		Sagatavo skatu jaunāko grupu, produktu un zemnieku attēlošanai dilstošā secībā. Skats: “home.blade.php” Modelis:

		“App\Group” “App\User” “App\Product”
farmers		Sagatavo skatu zemnieku attēlošanai dilstošā secībā. Skats: “farmers.blade.php” Modelis: “App\User”

2.5.1.4. *WelcomeController*

Metodes nosaukums	GET parametri	Apraksts
__construct		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas. Šim kontrolierim – tikai viesi.
index		Sagatavo skatu viesiem ar autentifikācijas un reģistrācijas formu. Skats: “welcome.blade.php”

2.5.1.5. *UserController*

Klase nodrošina *Zemnieku tirgus sistēmas* lietotājiem profila skatus un iespēju rediģēt savu lietotāja profilu.

Metodes nosaukums	GET parametri	Apraksts
__construct		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas. Šim kontrolierim – tikai reģistrēti lietotāji.
show	int \$id – lietotāja ID	Nodrošina lietotāja profila skata izveidi ar reģistrācijā sniegto informāciju.

		Skats: “user/profile.blade.php” Modelis: “App\User”
edit	int \$id – lietotāja ID	Nodrošina lietotāja profila datu rediģēšanas skata izveidi ar lietotāja datiem. Skats: “user/edit.blade.php” Modelis: “App\User”
update	int \$id – lietotāja ID	Metode ir atbildīga par lietotāja veikto izmaiņu saglabāšanu datubāzē. Modelis: “App\User”

2.5.1.6. *GroupController*

Klase nodrošina iespēju lietotājiem *Zemnieku tirgus sistēmā* veidot grupas un veikt atsevišķas darbības, kas pieejamas tikai lietotājiem, kuri ir grupā.

Metodes nosaukums	GET parametri	Apraksts
__construct		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas. Šim kontrolierim – tikai reģistrēti lietotāji.
index		Sagatavo skatu grupu saraksta attēlošanai dilstošā secībā, sākot ar jaunāko. Skats: “groups/index.blade.php” Modelis: “App\Group”
create		Nodrošina iespēju sistēmā izveidot jaunu grupas vienību.

		Skats: “groups/create.blade.php”
show	int \$id – grupas ID	Sagatavo skatu grupas profila attēlošanai. Skats: “groups/show.blade.php” Modelis: “App\Group” “App\User”
edit	int \$id – grupas ID	Nodrošina grupas datu rediģēšanas skata izveidi ar grupas datiem. Skats: ”groups/edit.blade.php”
destroy	int \$id – grupas ID	Metode izdzēš grupas ierakstu no datubāzes pēc lietotāja pieprasījuma. Modelis: “App\Group”
store		Metode ir atbildīga par jaunas grupas izveides pieprasījumu un saglabāšanu datubāzē. Modeļi: “App\Group” “App\User”
join	int \$id – grupas ID	Metode nodrošina lietotājiem iespēju pieteikt dalību grupā. Modeļi: “App\Group” “App\User”
leave	int \$id – grupas ID	Metode nodrošina lietotājiem iespēju atteikt dalību grupā. Modeļi:

		“App\Group” “App\User”
orders	int \$id – grupas ID	Sagatavo skatu ar neapstiprinātajiem grupas pasūtījumiem. Modeļi: “App\User” “App\Group” “App\Product” “App\Order”
postOrders		Metode ir atbildīga par neapstiprināto pasūtījumu apstiprināšanu un informācijas nosūtīšanu zemniekiem Modeļi: “App\User” “App\Group” “App\Product” “App\Order”

2.5.1.7. *ProductController*

Klase nodrošina Zemnieku tirgus sistēmas lietotājiem iespēju pievienot produktus un veikt dažādas datu manipulācijas.

Metodes nosaukums	GET parametri	Apraksts
__construct		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas. Šim kontrolierim – tikai reģistrēti lietotāji.
index		Sagatavo skatu produktu saraksta attēlošanai dilstošā secībā, sākot ar jaunāko. Skats: “product/index.blade.php”

		Modelis: “App\Product”
create		Nodrošina iespēju sistēmā izveidot jaunu produkta vienību. Skats: “product/create.blade.php” Modelis: “App\Category”
store		Metode ir atbildīga par jauna produkta izveides pieprasījumu un saglabāšanu datubāzē. Modelis: “App\Product”
show	int \$id – produkta ID	Sagatavo skatu produkta profila attēlošanai Skats: “product/show.blade.php” Modelis: “App\Product”
edit	int \$id – produkta ID	Nodrošina produkta datu rediģēšanas skata izveidi ar produkta datiem. Skats: “product/edit.blade.php”
update	int \$id – produkta ID	Metode nodrošina eksistējoša produkta ieraksta atjaunošanu datubāzē pēc pieprasījuma nosūtīšanas. Modelis: “App\Product”
destroy	int \$id – produkta ID	Metode izdzēs produkta ierakstu no datubāzes pēc lietotāja pieprasījuma. Modelis: “App\Product”

2.5.1.8. *CategoryController*

Metodes nosaukums	GET parametri	Apraksts
__construct		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas. Šim kontrolierim – tikai lietotājs ar tipu ‘Administrators’.
index		Sagatavo skatu kategoriju saraksta attēlošanai dilstošā secībā, sākot ar jaunāko. Skats: “category/index.blade.php” Modelis: “App\Category”
create		Nodrošina kategorijas datu pievienošanas skatu. Skats: “category/create.blade.php” Modelis: “App\Category”
edit	int \$id – kategorijas ID	Nodrošina kategorijas datu rediģēšanas skata izveidi ar kategorijas datiem. Skats: “category/edit.blade.php”
update	int \$id – kategorijas ID	Metode ir atbildīga par administratora veikto izmaiņu saglabāšanu datubāzē. Modelis: ”App\Category”
destroy		Metode izdzēš kategorijas ierakstu no datubāzes pēc administratora pieprasījuma. Modelis: “App\Category”

2.5.1.9. OrderController

Metodes nosaukums		Apraksts
__construct		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas. Šim kontrolierim – tikai reģistrēti lietotāji.
getCart		Nodrošina produktu grozā ievietoto produktu skatu. Skats: "Order/cart.blade.php"
getCartAdd	int \$id – produkta ID	Metode ir atbildīga par produktu pievienošanu produktu grozam.
getCartRemove	int \$id – produkta ID	Metode ir atbildīga par produktu izņemšanu no produktu groza.
getPlace	int \$id – produkta ID	Nodrošina noformētā pasūtījuma detaļu skatu. Skats: "Order/place.blade.php"
postPlace		Metode ir atbildīga par produktu grozā esošo produktu pievienošanu datu bāzei. Modelis: "App\Order"

2.5.2. Skati

Visas datnes atrodas sistēmas direktorijā: *resources/views*

Datne	Mērķis
<i>auth/login.blade.php</i>	Lietotāja pieteikšanās skats.
<i>auth/password.blade.php</i>	Lietotāja pazaudētas paroles maiņas pieprasījuma skats.

<i>auth/register.blade.php</i>	Lietotāja reģistrēšanās skats.
<i>auth/reset.blade.php</i>	Lietotāja paroles maiņas skats.
<i>emails/password.blade.php</i>	Lietotāja paroles atjaunošanas skats.
<i>errors/503.blade.php</i>	Kļūdas '503' skats – pakalpojums nav pieejams.
<i>groups/index.blade.php</i>	Visas grupas, grupēts no jaunākās skats.
<i>groups/create.blade.php</i>	Grupās izveides skats.
<i>groups/show.blade.php</i>	Grupās detalizēta apraksta skats.
<i>groups/edit.blade.php</i>	Grupās rediģēšanas skats.
<i>groups/orders.blade.php</i>	Grupās pasūtījumu skats.
<i>product/index.blade.php</i>	Visi produkti, grupēti no jaunākā skats.
<i>product/create.blade.php</i>	Produktu izveides skats.
<i>product/show.blade.php</i>	Produkta detalizēta apraksta skats.
<i>product/edit.blade.php</i>	Produkta rediģēšanas skats.
<i>user/profile.blade.php</i>	Lietotāja profila skats.
<i>user/edit.blade.php</i>	Lietotāja profila labošanas skats.
<i>order/cart.blade.php</i>	Lietotāja produktu groza skats.
<i>order/make.blade.php</i>	Grupās neapstiprināto pasūtījumu skats.
<i>order/place.blade.php</i>	Lietotāja pasūtījuma apstiprinājuma skats.
<i>app.blade.php</i>	Navigācijas skats.
<i>farmers.blade.php</i>	Zemnieku informācijas skats.
<i>home.blade.php</i>	Sāku lapas skats dalībniekiem.
<i>welcome.blade.php</i>	Sāku lapas skats viesiem.

3. TESTĒŠANAS DOKUMENTĀCIJA

3.1. Testēšanas procesa apraksts

Pēc katras Zemnieku tirgus sistēmas funkcijas izveides tika veikta šīs funkcijas vienībtestēšana. Testēšana tika veikta, lai pēc funkcijas izveides tiktu atrastas ar to saistītās kļūdas un tās tiktu izlabotas, lai izvairītos no neparedzētas sistēmas darbības, kuras pamatā varētu būt nepilnīgi izstrādātas funkcijas.

3.2. Vienībtestēšanas rezultāti

3.2.1. Kontrolieris *WelcomeController*

3.2.1.1. *__construct*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas.

3.2.1.2. *index*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode		Tiek inicializēts skats "welcome.blade.php". Attēlota reģistrācijas un iežurnalēšanās forma viesiem.

3.2.2. Kontrolieris *HomeController*

3.2.2.1. *__construct*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas.

3.2.2.2. *index*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode		Tiek inicializēts skats "home.blade.php". Attēlota lapa, kas satur informāciju par jaunākajiem zemniekiem, produktiem un grupām.

3.2.2.3. *farmers*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
-----	----------	-----------	-----------------------

Tiek izsaukta metode		Tiek inicializēts skats “farmers.blade.php”. Attēlota lapa ar zemniekiem dīlstošā secībā.
----------------------	--	---

3.2.3. Kontrolieris UserController

3.2.3.1. __construct

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas.

3.2.3.2. show

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Tiek inicializēts skats “user/profile.blade.php”. Attēlota lietotāja profila informācija, kas satur reģistrācijas procesā ievadītos datus.

3.2.3.3. edit

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode		Tiek inicializēts skats “user/edit.blade.php”. Lauki, kas datubāzē ir ar vērtību ‘NULL’ tiek attēloti kā tukši. Pārējie lauki tiek aizpildīti ar lietotāja datiem.
	Pārbaude vai lietotājs ir autorizēts šīs funkcijas izsaušanai.		Ja lietotājs ir autorizēts veikt šo darbību, tad tiek atgriezts skats, ja nē, lietotājs tiek pāradresēts uz sākuma lapu ar ziņu “Jums nav tiesību veikt šādu darbību!”
	Izvada pareizības pārbaude.		Lietotāja profila datu izvades laukos tiek izvadīts:

			<p>“Vārds: <i>name</i>”</p> <p>“Uzvārds: <i>surname</i>”</p> <p>“E-pasts: <i>email</i>”</p> <p>“Telefons: <i>telephone</i>”</p> <p>Zemnieka profila datu izvades laukos tiek izvadīts:</p> <p>“Vārds: <i>name</i>”</p> <p>“Uzvārds: <i>surname</i>”</p> <p>“Z/S nosaukums: <i>title</i>”</p> <p>“Z/S apraksts: <i>description</i>”</p> <p>“Adrese: <i>location</i>”</p>
--	--	--	---

3.2.3.4. *update*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Obligāto datu lauku pārbaude lietotāja profilam.	Visi formas datu ievades lauki tiek atstāti tukši.	<p>Paziņojumi:</p> <p>“Lauks ‘name’ ir obligāts.”</p> <p>“Lauks ‘surname’ ir obligāts.”</p> <p>“Lauks ‘telephone’ ir obligāts.”</p> <p>“Lauks ‘email’ ir obligāts.”</p>
	Obligāto datu lauku pārbaude zemnieka profilam	Visi formas datu ievades lauki tiek atstāti tukši.	<p>“Lauks ‘name’ ir obligāts.”</p> <p>“Lauks ‘title’ ir obligāts.”</p> <p>“Lauks ‘description’ ir obligāts.”</p> <p>“Lauks ‘location’ ir obligāts.”</p>
	Obligāto datu lauku minimālā garuma ierobežojuma pārbaude.	Ievadītie dati neatbilst minimālā garuma prasībām.	Visiem laukiem jāsaturs vismaz 3 simbolu garums.
	Datu lauku maksimālā garuma ierobežojuma pārbaude.	Ievadītie dati pārsniedz maksimālā garuma prasības.	<p>“Lauks ‘name’ nedrīkst pārsniegt 30 simbolus.”</p> <p>“Lauks ‘surname’ nedrīkst pārsniegt 30 simbolus.”</p> <p>“Lauks ‘title’ nedrīkst pārsniegt 30 simbolus.”</p>

		<p>“Lauks ‘description’ nedrīkst pārsniegt 254 simbolus.”</p> <p>“Lauks ‘location’ nedrīkst pārsniegt 254 simbolus.”</p>
	Pareizu ievaddatu apstrāde.	Ievada datubāzē visus aizpildītos laukus.

3.2.3.5. *destroy*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Metode izdzēš lietotāja ierakstu un visas ar to saistītās saites.

3.2.4. *Kontrolieris GroupController*

3.2.4.1. *__construct*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas.

3.2.4.2. *index*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Tiek inicializēts skats “groups/index.blade.php” ar jaunākajiem grupas ierakstiem.

3.2.4.3. *create*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Tiek inicializēts skats “groups/create.blade.php” ar tukšiem ievades laukiem.

Pārbaude vai lietotājs neatrodas grupa vai nav kādas grupas veidotājs.		Ja kāds no nosacījumiem ir patiess, izvada kļūdas paziņojumu, pretēji, izpilda funkciju.
--	--	--

3.2.4.4. *store*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Obligāto datu lauku pārbaude grupas ierakstam.	Visi formas datu ievades lauki tiek atstāti tukši.	Paziņojumi: “Lauks ‘Title’ ir obligāts.” “Lauks ‘Description’ ir obligāts.” “Lauks ‘Location’ ir obligāts.”
	Datu lauku minimālā garuma ierobežojuma pārbaude.	Ievadītie dati neatbilst minimālā garuma prasībām.	Paziņojumi: “Lauks ‘Title’ satur vismaz 3 simbolus.” “Lauks ‘Description’ satur vismaz 3 simbolus.” “Lauks ‘Location’ satur vismaz 3 simbolus.”
	Datu lauku maksimālā garuma ierobežojuma pārbaude.	Ievadītie dati pārsniedz maksimālā garuma prasības.	Paziņojumi: “Lauks ‘Title’ nepārsniedz 30 simbolus.” “Lauks ‘Description’ nepārsniedz 254 simbolus.” “Lauks ‘Location’ nepārsniedz 254 simbolus.”
	Pareizu ievaddatu apstrāde.		Ievada datubāzē visus aizpildītos laukus.

3.2.4.5. *show*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
-----	----------	-----------	-----------------------

Tiek izsaukta metode.		Tiek inicializēts skats “groups/show.blade.php”. Attēlota grupas informācija, kas satur izveides procesā ievadītos datus.
-----------------------	--	---

3.2.4.6. *edit*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Tiek inicializēts skats “group/edit.blade.php”.
	Pārbaude vai lietotājs ir autorizēts šīs funkcijas izsaušanai.		Ja lietotājs ir autorizēts veikt šo darbību, tad tiek atgriezts skats, ja nē, lietotājs tiek pārdresēts uz sākuma lapu ar ziņu “Jums nav tiesību veikt šādu darbību!”
	Izvada pareizības pārbaude.		Grupās datu izvades laukos tiek izvadīts: “Nosaukums: title” “Apraksts: description” “Adrese: location”

3.2.4.7. *update*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Obligāto datu lauku pārbaude grupas ierakstam.	Visi formas datu ievades lauki tiek atstāti tukši.	Paziņojumi: “Lauks ‘Title’ ir obligāts.” “Lauks ‘Description’ ir obligāts.” “Lauks ‘Location’ ir obligāts.”
	Datu lauku minimālā garuma ierobežojuma pārbaude.	Ievadītie dati neatbilst minimālā garuma prasībām.	Paziņojumi: “Lauks ‘Title’ satur vismaz 3 simbolus.” “Lauks ‘Description’ satur vismaz 3 simbolus.”

			“Lauks ‘Location’ satur vismaz 3 simbolus.”
	Datu lauku maksimālā garuma ierobežojuma pārbaude.	Ievadītie dati pārsniedz maksimālā garuma prasības.	Paziņojumi: “Lauks ‘Title’ nepārsniedz 30 simbolus.” “Lauks ‘Description’ nepārsniedz 254 simbolus.” “Lauks ‘Location’ nepārsniedz 254 simbolus.”
	Pareizu ievaddatu apstrāde.		Ievada datubāzē visus aizpildītos laukus.

3.2.4.8. *join*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Pārbaude vai lietotājs jau neatrodas kādā grupā.		Ja lietotājs ir kādā grupā, izvada kļūdas paziņojumu, pretēji, izpilda funkciju.

3.2.4.9. *leave*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Pārbaude vai lietotājs atrodas grupā		Ja lietotājs ir grupā, dzēš saiti starp lietotāju un grupu.

3.2.4.10. *orders*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Pārbaude grupā ir izveidoti pasūtījumi.		Ja grupā ir izveidoti pasūtījumi, inicializē skatu, ja nav, atgriež kļūdas paziņojumu.

3.2.4.11. *postOrders*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Saņemts apstiprinājums izpildīt pasūtījumus.		Atzīmē pasūtījumu statusu kā 'true' un nosūta e-pastu par pasūtījuma detaļām zemniekiem.

3.2.4.12. *destroy*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Metode izdzēš grupas ierakstu un visas ar to saistītās saites.

3.2.5. *Kontrolieris ProductController*

3.2.5.1. *__construct*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas.

3.2.5.2. *index*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Tiek inicializēts skats "product/index.blade.php" ar jaunākajiem produktu ierakstiem.

3.2.5.3. *create*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Tiek inicializēts skats "product/create.blade.php".
	Pārbaude vai lietotājs ir autorizēts šīs		Ja lietotāja tips ir "zemnieks" izpilda funkciju, pretēji, tiek izvadīts kļūdas paziņojums.

funkcijas izsaukšanai.		
---------------------------	--	--

3.2.5.4. *store*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Obligāto datu lauku pārbaude produkta ierakstam.	Visi formas datu ievades lauki tiek atstāti tukši.	Paziņojumi: “Lauks ‘name’ ir obligāts.” “Lauks ‘description’ ir obligāts.” “Lauks ‘category’ ir obligāts.” “Lauks ‘price’ ir obligāts.”
	Datu lauku minimālā garuma ierobežojuma pārbaude.	Ievadītie dati neatbilst minimālā garuma prasībām.	Paziņojumi: “Lauks ‘name’ satur vismaz 3 simbolus.” “Lauks ‘description’ satur vismaz 3 simbolus.”
	Datu lauku maksimālā garuma ierobežojuma pārbaude.	Ievadītie dati pārsniedz maksimālā garuma prasības.	Paziņojumi: “Lauks ‘title’ nepārsniedz 30 simbolus.” “Lauks ‘description’ nepārsniedz 254 simbolus.”
	Pareizu ievaddatu apstrāde.		Ievada datubāzē visus aizpildītos laukus.

3.2.5.5. *edit*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Tiek inicializēts skats “product/edit.blade.php”
	Pārbaude vai lietotājs ir autorizēts šīs funkcijas izsaukšanai.		Ja lietotājs ir autorizēts veikt šo darbību, tad tiek atgriezts skats, ja nē, lietotājs tiek pāradresēts uz sākuma lapu ar ziņu “Jums nav tiesību veikt šādu darbību!”

Izvada pareizības pārbaude.		Produkta datu izvades laukos tiek izvadīts: “Nosaukums: <i>name</i> ” “Apraksts: <i>description</i> ” “Bioloģisks: <i>bio</i> ” “Kategorija: <i>category_id</i> ” “Cena: <i>price</i> ”
-----------------------------	--	--

3.2.5.6. *show*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Tiek inicializēts skats “product/show.blade.php”. Attēlota produkta informācija, kas satur izveides procesā ievadītos datus.

3.2.5.7. *update*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Obligāto datu lauku pārbaude grupas ierakstam.	Visi formas datu ievades lauki tiek atstāti tukši.	Paziņojumi: “Lauks ‘Name’ ir obligāts.” “Lauks ‘Description’ ir obligāts.” “Lauks ‘Price’ ir obligāts.”
	Datu lauku minimālā garuma ierobežojuma pārbaude.	Ievadītie dati neatbilst minimālā garuma prasībām.	Paziņojumi: “Lauks ‘Name’ satur vismaz 3 simbolus.” “Lauks ‘Description’ satur vismaz 3 simbolus.” “Lauks ‘Price’ ir decimāls skaitlis”
	Datu lauku maksimālā garuma ierobežojuma pārbaude.	Ievadītie dati pārsniedz maksimālā garuma prasības.	Paziņojumi: “Lauks ‘Title’ nepārsniedz 30 simbolus.” “Lauks ‘Description’ nepārsniedz 254 simbolus.”

	Pareizu ievaddatu apstrāde.		Ievada datubāzē visus aizpildītos laukus.
--	-----------------------------	--	---

3.2.5.8. *destroy*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Metode izdzēš produkta ierakstu un visas ar to saistītās saites.

3.2.6. *Kontrolieris CategoryController*

3.2.6.1. *__construct*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas.

3.2.6.2. *index*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Tiek inicializēts skats “category/index.blade.php” ar jaunākajiem kategoriju ierakstiem.

3.2.6.3. *create*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Tiek inicializēts skats “category/create.blade.php”.
	Pārbaude vai lietotājs ir autorizēts šīs funkcijas izsaušanai.		Ja lietotāja tips ir “Administrators” izpilda funkciju, pretēji, tiek izvadīts kļūdas paziņojums.

3.2.6.4. *store*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Obligāto datu lauku pārbaude produkta ierakstam.	Visi formas datu ievades lauki tiek atstāti tukši.	Paziņojumi: “Lauks ‘name’ ir obligāts.”
	Datu lauku minimālā garuma ierobežojuma pārbaude.	Ievadītie dati neatbilst minimālā garuma prasībām.	Paziņojumi: “Lauks ‘name’ satur vismaz 3 simbolus.”
	Datu lauku maksimālā garuma ierobežojuma pārbaude.	Ievadītie dati pārsniedz maksimālā garuma prasības.	Paziņojumi: “Lauks ‘name’ nepārsniedz 30 simbolus.”
	Pareizu ievaddatu apstrāde.		Ievada datubāzē visus aizpildītos laukus.

3.2.6.5. *edit*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Tiek inicializēts skats “category/edit.blade.php”
	Pārbaude vai lietotājs ir autorizēts šīs funkcijas izsaukšanai.		Ja lietotājs tips ir “Administrators”, tad tiek atgriezts skats, ja nē, lietotājs tiek pārdresēts uz sākuma lapu ar ziņu “Jums nav tiesību veikt šādu darbību!”
	Izvada pareizības pārbaude.		Produkta datu izvades laukos tiek izvadīts: “Nosaukums: <i>name</i> ”

3.2.6.6. *update*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
-----	----------	-----------	-----------------------

Obligāto datu lauku pārbaude grupas ierakstam.	Visi formas datu ievades lauki tiek atstāti tukši.	Paziņojumi: “Lauks ‘Name’ ir obligāts.”
Datu lauku minimālā garuma ierobežojuma pārbaude.	Ievadītie dati neatbilst minimālā garuma prasībām.	Paziņojumi: “Lauks ‘Name’ satur vismaz 3 simbolus.”
Datu lauku maksimālā garuma ierobežojuma pārbaude.	Ievadītie dati pārsniedz maksimālā garuma prasības.	Paziņojumi: “Lauks ‘Name’ nepārsniedz 30 simbolus.”
Pareizu ievaddatu apstrāde.		Ievada datubāzē visus aizpildītos laukus.

3.2.6.7. *destroy*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Metode izdzēš kategorijas ierakstu un visas ar to saistītās saites.

3.2.7. *Kontrolieris OrderController*

3.2.7.1. *__construct*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode		Definē dalībnieku grupas, kam pieejamas attiecīgās funkcijas.

3.2.7.2. *getCart*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.		Tiek inicializēts skats “order/cart.blade.php” ar lietotāja pievienotajiem produktiem grozam.

3.2.7.3. *getCartadd*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.	int \$product_id	Ja iepirkuma grozā jau ir produkts, palielināt daudzumu par 1, pretēji, ielikt grozā šo produktu.

3.2.7.4. *getCartremove*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.	int \$product_id	Samazina groza produkta daudzumu par 1 vienību.

3.2.7.5. *getPlace*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Tiek izsaukta metode.	Saraksts ar lietotāja, produkta, pasūtījuma, grupas informāciju.	Tiek inicializēts skats "order/make.blade.php". Parāda visus grupas pasūtījumus, kuri nav izpildīti.

3.2.7.6. *postPlace*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
	Pareizu ievaddatu apstrāde.	Saraksts ar lietotāja, produkta, pasūtījuma, grupas informāciju.	Atzīmē datubāzē pasūtījumu statusu kā 'true' un nosūta e-pastu zemniekiem par pasūtījuma detaļām.

3.2.7.7. *destroy*

Nr.	Apraksts	Ievaddati	Sagaidāmais rezultāts
-----	----------	-----------	-----------------------

Tiek izsaukta metode.	Metode izdzēs pasūtījuma ierakstu un visas ar to saistītās saites.
-----------------------	--

4. PROJEKTA ORGANIZĀCIJA

Uzsākot projektu, tikai veiktas konsultācijas ar kvalifikācijas darba vadītāju un prakses vadītāju, kuru rezultātā tika iegūts priekšstats par nepieciešamo funkcionalitāti, prasībām un ierobežojumiem, kā arī stilu, kādu vajadzētu ieturēt darba izstrādes procesā. Tika uzsākta programmas projektējuma specifiku un programmatūras prasību apraksta plānošana un izveide, kurai sekoja tālākas konsultācijas par paveikto un par nepieciešamajiem sagatavošanās darbiem pirms programmēšanas uzsākšanas.

Tīmekļa bāzētās sistēmas izstrādē izmantota programmēšanas valoda PHP 5 un tās darba rāmis *Laravel 5*, kā arī relāciju datubāzes vadības sistēma MySQL. Aplikācijas izstrādes process noritēja pēc MVC arhitektūras standarta. Sistēmas programmēšana tika veikta izstrādes vidē *JetBrains PhpStorm 8.0.3*, kurai ir speciāls atbalsts izstrādājot *Laravel 5* projektus.

Lai ieturētu programmēšanas labo praksi programmatūras koda plānošanai un rakstīšana tika pētīta *Laravel 5* oficiālā dokumentācija [1] un uz to balstīts programmatūras kods.

5. KONFIGURĀCIJAS PĀRVALDĪBA

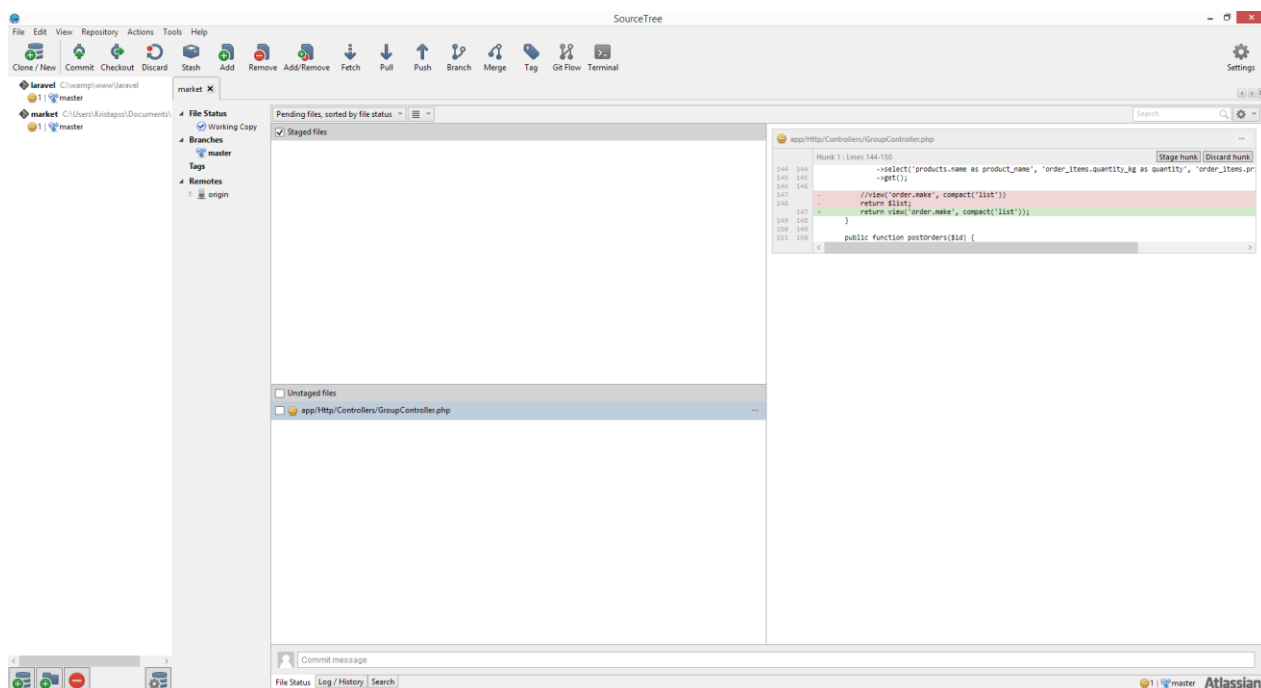
Lai nodrošinātu konfigurācijas pārvaldību, tika izmantota versiju kontroles sistēma *git* [4]. Versiju kontroles mērķis projektā, galvenokārt, ir reģistrēt izmaiņas pirmkoda komponentēs un piekļūt sistēmas iepriekš izveidotām revīzijām, kas ir nodrošinātas, ja tiek atklātas būtiskas kļūdas kādā no jaunākajām revīzijām un ir nepieciešams atgriezties kādā iepriekšējā sistēmas stāvoklī.

Versiju vadība īstenota izmantojot Github, grafiskās saskarnes Web aplikācijas, repozitoriju, kas tika pārvaldīta ar *SourceTree* [5] klientu.

Github repozitorija pieejama: <https://github.com/KristapsBerzins/market>

Neatkarīgi no vietas un laika, kur tika uzsākta vai turpināta sistēmas izstrāde mainot darba vidi, uzsākot darbu, tika iegūta revīzija, kurai tika veiktas pēdējās izmaiņas. Revīzijas tiek reģistrētas automātiski un saglabātas mākonserverī.

Bildē zemāk redzams *SourceTree* revīziju pārvaldības rīks.



6. DARBIETILPĪBAS NOVĒRTĒJUMS

6.1. Sākotnējais darbietilpības novērtējums

Sākotnējais darbietilpības novērtējums tika izdarīts balstoties uz izvirzītajām projekta prasībām, sistēmas izstrādātāja praktisko pieredzi un prakses vadītāja subjektīvo viedokli, ņemot vērā sistēmas izstrādātāja esošās zināšanas, pieredzi darbā ar līdzīgiem projektiem. Balstoties uz šiem pieņēmumiem sistēmas izstrādātājs, konsultējoties ar prakses vadītāju, izveidoja aptuveno izstrādājamās sistēmas funkcionalitāti, kuru sistēmas izstrādātājs spētu izstrādāt, iekļaujoties trīs personmēnešu termiņā.

6.2. Darbietilpības novērtējums pēc sistēmas realizācijas

Tiek pieņemts, ka 1 persondienas raksturojošais lielums ir 8 stundas. Jāņem vērā, ka tabulā rakstītie patērēto dienu skaita lielumi ir aptuveni un uzverami informatīvi, tomēr ir pietiekami objektīvi, lai tos izmantotu turpmāku projektu plānošanā un darbietilpības novērtēšanas procesā. Patērēto dienu uzskaitē tika īstenota pašrocīgi, ievadot datus Microsoft Excel lietojumprogrammā katras darba nedēļas noslēgumā. Kontrolieru programmēšanā iekļauts laiks skatu izveidei attiecīgajām kontrolieru metodēm.

Projekta posms	Paveiktais darbs	Dienas
Sistēmas analīze	Funkciju definēšana	3
	Datu bāzes projektēšana	2
Dokumentācijas izstrāde	Konsultācijas ar prakses vadītāju	3
	Pamatprasību definēšana	1
	PPS izstrāde	7
Projektēšana	Moduļu projektēšana	3
	PPA izstrāde	1
Programmēšana	Izstrādes ietvara konfigurēšana	2
	Sistēmas tīmekļa dizaina projektēšana	10
	Funkcionālo klašu programmēšana	
	Datu bāzes migrāciju izveide	2
	Model_User	0.5
	Model_UserType	0.5
	Model_Group	0.5

	Model_Product	0.5
	Model_Order	0.5
	Model_OrderItem	0.5
	Model_Category	0.5
	Controller_Home	1
	Controller_Welcome	1
	Controller_User	3
	Controller_Group	4
	Controller_Product	2
	Controller_Order	4
	Controller_Category	2
Testēšana	Sistēmas vienībtestu plānošana, dokumentēšana un funkciju testēšana	
	Testēšanas dokumentācijas izveide	2
	Test_WelcomeController	0.5
	Test_HomeController	0.5
	Test_UserController	1
	Test_GroupController	1
	Test_ProductController	0.5
	Test_CategoryController	0.5
	Test_OrderController	1
	Kļūdu un funkcionalitātes nepilnību novēršana	12
	Sistēmas akceptēšana	2
Ieviešana	Ieviešana	3

Pēc šīs darbietilpības noteikšanas metodes kopējais projektam nepieciešamo persondienu skaits ir 78,5 jeb 628 stundas. Vienu personmēnesi veido 22 persondienas jeb 176 stundas, tātad kopējais projektam nepieciešamo personmēnešu skaits ir 3,5.

Programmēšanai nepieciešamais persondienu skaits ir 34 jeb 272 stundas, tātad kopējais programmēšanai nepieciešamo personmēnešu skaits ir 1,5.

7. KVALITĀTES NODROŠINĀŠANA

Lai nodrošinātu izstrādājamās sistēmas kvalitāti, izstrādes procesā tika veikti vairāki pasākumi:

- Laravel 5 dokumentācijas izpēte, zināšanu pilnveidošana un apguve kvalitatīvas sistēmas izgatavošanai;
- Sistēmas testēšana, lai atklātu un izlabotu kļūdas sistēmā;
- Projekta dokumentācijas izstrāde, izmantojot atbilstošus nozares standartus;
- Versiju kontroles rīka *git* izmantošana sistēmas relīžu definēšanai un glabāšanai.
- Konsultācijas ar prakses vadītāju un kvalifikācijas darba vadītāju, lai pārliecinātos, ka darba process tiek organizēts atbilstoši nepieciešamām prasībām.
- Dokumentācijas izstrāde pēc labās prakses un standartiem izmantojot gan LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai”, gan LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis”.

NOBEIGUMS

Zemnieku tirgus sistēma tika sekmīgi izstrādāta atbilstoši tās prasību specifikācijai, nedaudz pārsniedzot sākotnēji noteikto laika limitu.

Darba izstrādes gaitā tika apgūti projekta pārvaldības un dokumentu izstrādes pamati. Kā grūtāko uzdevumu sistēmas izstrādes procesā var minēt projekta prasību projektēšanu un plānošanu pirms programmēšanas uzsākšanas, kā arī funkciju programmēšanu atbilstoši izstrādātajām prasībām.

Jāatzīst, ka tika iegūta vērtīga pieredze izstrādājot sistēmu satvarā Laravel 5, ar kuru iepriekš ir bijusi minimāla pieredze. Lai gan sistēma funkcionē pilnvērtīgi, vēl joprojām tiek turpināti darbi pie sistēmas funkcionalitātes un vizuālā izskata uzlabošanas.

Nākotnē ar sistēmu plānots veikt sekojošas darbības:

- Novērot un uzlabot sistēmas ātrdarbību.
- Papildināt funkcionalitāti ar jaunām funkcijām.
- Koriģēt programmatūras kodu, lai panāktu pēc iespējas efektīvāku risinājumu.

IZMANTOTĀ LITERATŪRA

Tīmekļa resursi

- [1] – Laravel 5 documentation. [tiešsaiste] [atsauce 06.04.2015] Pieejams internetā:
<http://laravel.com/docs/5.0>
- [2] – Bootstrap CSS. [tiešsaiste] [atsauce 06.04.2015] Pieejams internetā:
<http://getbootstrap.com/css>
- [3] – Laravel 5 Fundamentals [tiešsaiste] [atsauce 06.04.2015] Pieejams internetā:
<https://laracasts.com/series/laravel-5-fundamentals>
- [4] – Git [tiešsaiste] [atsauce 31.05.2015] Pieejams internetā: <https://git-scm.com/doc>
- [5] – SourceTree [tiešsaiste] [atsauce 31.05.2015] Pieejams internetā:
<https://www.sourcetreeapp.com/>
- [8] – RFC 5322 standarts [tiešsaiste] [atsauce 31.05.2015] Pieejams internetā:
<https://tools.ietf.org/html/rfc5322>

Latvijas Valsts standarti

- [6] - LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai”.
- [7] –LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis”.

PIELIKUMI

Zemnieku tirgus sistēmas pirmkoda fragmenti

Tā kā sistēma izstrādāta izmantojot MVC satvaru, zemāk tiks atspoguļoti pirmkodi no kontroliera, modeļa un skata.

Kontrolieris

Datne App\Controllers\GroupController.php

```
<?php namespace App\Http\Controllers;

use App\Http\Requests;
use App\Http\Controllers\Controller;
use App\Group;
use App\Http\Requests\CreateGroupRequest;
use Illuminate\Http\Request;
use Input;
use Auth;
use App\User;
use DB;
use App\Order;
use App\Product;
use App\OrderItem;

class GroupController extends Controller {

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('auth');
    }

    /**
     * Display a listing of the resource.
     *
     * @return Response
     */
    public function index()
    {
        $groups = Group::latest()->get();

        return view('groups.index', compact('groups'));
    }
}
```

```

/**
 * Show the form for creating a new resource.
 *
 * @return Response
 */
public function create()
{
    if (Auth::guest())
    {
        return redirect('welcome');
    }

    return view('groups.create');
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return Response
 */
public function show($id)
{
    $group = Group::findOrFail($id);
    $usergroup = Auth::User()->group_id;
    $userslist = DB::table('users')->where('group_id', $id)-
>get();

    return
view('groups.show', compact('group', 'userslist', 'usergroup'));
}

/**
 * Save a new Group
 *
 * @param CreateGroupRequest $request
 * @return
\Illuminate\Http\RedirectResponse|\Illuminate\Routing\Redirector
 */
public function store(CreateGroupRequest $request)
{
    $group = new Group($request->all());
    $group->title = Input::get('title');
    $group->description = Input::get('description');
    $group->location = Input::get('location');
    $group->owner_id = Auth::User()->id;
    $group->save();

    $user = Auth::User()->id;

    DB::table('users')->where('id', '=', $group->owner_id)-
>update(array('group_id' => $group->id));

    return redirect('groups');
}

```

```

    }

    /**
     * Join the specified group
     *
     * @param int $id
     * @return Response
     */
    public function join($id)
    {
        $user = Auth::User()->id;
        $group = Group::findOrFail($id);

        DB::table('users')->where('id', '=', $user)-
>update(array('group_id' => $group->id));

        return redirect('groups/'.$id)->with('message', 'Jūs esat
pievienojies grupai!');
    }

    /**
     * Leave the specified group
     *
     * @param int $id
     * @return Response
     */
    public function leave($id) {
        $user = Auth::User()->id;
        $group = Group::findOrFail($id);

        if($user == $group->owner_id){
            return redirect('groups/'.$id)->with('message', 'Grupas
īpašnieks nevar pamest grupu!');
        }
        else {
            DB::table('users')->where('id', '=', $user)-
>update(array('group_id' => NULL));

            return redirect('groups/'.$id)->with('message', 'Jūs esat
pametis grupu!');
        }
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param int $id
     * @return Response
     */
    public function edit($id) {
        $group = Group::findOrFail($id);
        if(Auth::User()->id == $group->owner_id) {
            return view('groups.edit', compact('group'));
        }
        else {
            return redirect('home')->with('message', 'Jūs nav

```

```

tiesību veikt šādu darbību!');
    }
}

public function update($id, CreateGroupRequest $request) {
    $group = Group::findOrFail($id);

    $group->update($request->all());

    return redirect('groups/' . $id) ->with('message', 'Grupa
veiksmīgi atjaunota!');
}

/**
 * Show the list of uncompleted orders made by group.
 *
 * @param int $id
 * @return Response
 */
public function orders($id) {
    $group = Group::findOrFail($id);

    $list = DB::table('groups')
        ->join('orders', 'orders.group_id', '=', 'groups.id')
        -
    >join('order_items', 'order_items.order_id', '=', 'orders.id')
        -
    >join('products', 'products.id', '=', 'order_items.product_id')
        ->join('users', 'users.id', '=', 'orders.user_id')
        ->where('status', '=', false)
        ->where('groups.id', '=', $id)
        ->select('products.name as
product_name', 'order_items.quantity_kg as
quantity', 'order_items.price as price', 'users.title as name')
        ->get();

    return view('order.make', compact('list'));
}

/**
 * Complete the group order.
 *
 * @param int $id
 * @return Response
 */
public function postOrders($id) {
    $group = Group::findOrFail($id);

    $list = DB::table('groups')
        ->join('orders', 'orders.group_id', '=', 'groups.id')
        -
    >join('order_items', 'order_items.order_id', '=', 'orders.id')
        -
    >join('products', 'products.id', '=', 'order_items.product_id')
        ->join('users', 'users.id', '=', 'orders.user_id')
        ->where('status', '=', false)
        ->where('groups.id', '=', $id)

```

```

        ->select('products.name as
product_name', 'order_items.quantity_kg as
quantity', 'order_items.price as price', 'users.title as name')
        ->get();

        // email function

        return redirect('groups/'.$id)->with('message', 'Pasūtījums
veiksmīgi noformēts!');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return Response
     */
    public function destroy($id) {
        $group = Group::findOrFail($id);
        $user = Auth::User()->id;

        if($user == $group->owner_id) {
            $group->delete();

            return redirect('groups.index')->with('message', 'Grupa
izdzēsta veiksmīgi!');
        }
        else {
            return redirect('groups/'.$id)->with('message', 'Jums nav
tiesību veikt šādu darbību.');
```

Modelis

Datne App\Group.php

```

<?php namespace App;

use Illuminate\Database\Eloquent\Model;

class Group extends Model {

    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'groups';

    /**
     * The attributes that are mass assignable.
```

```

    *
    * @var array
    */
    protected $fillable = [
        'title',
        'description',
        'location',
        'owner_id'
    ];

    public function users() {
        return $this->hasMany('App\User');
    }

    public function owner() {
        return $this->belongsTo('App\User');
    }

    public function orders() {
        return $this->hasMany('App\Order');
    }
}

```

Skats

Datne Resources\Views\Groups\show.blade.php

```

@extends('app')

@section('content')
    @if (Session::has('message'))
        <div class="alert alert-success"> {{ Session::get('message') }}</div>
    @endif
    <h1>{{ $group->title }}</h1>
    <article>
        {{ $group->description }}
        {{ $group->location }}
    </article>
    <table class="table table-striped">
        <thead>
            <tr>
                <th>Firstname</th>
                <th>Lastname</th>
                <th>Email</th>
            </tr>
        </thead>

```

```

  @foreach($userslist as $userlist)
    <tr>
      <td>{{ $userlist->name }}</td>
      <td>{{ $userlist->surname }}</td>
      <td>{{ $userlist->email }}</td>
    </tr>
  @endforeach
</tbody>
</table>
@if($usergroup == NULL)
  {!! Form::open(array('action' => array('GroupController@join',
  $group->id))) !!}
  {!! Form::submit('Pievienoties grupai', ['class' => 'btn btn-
  primary btn-sm']) !!}
  {!! Form::close() !!}
@else
  {!! Form::open(array('action' => array('GroupController@leave',
  $group->id))) !!}
  {!! Form::submit('Pamest Grupu', ['class' => 'btn btn-danger
  btn-sm']) !!}
  {!! Form::close() !!}
@endif
@endsection

```

Kvalifikācijas darbs „**Zemnieku tirgus sistēma**” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: **Kristaps Bērziņš** _____ .05.2015.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: **doc. Edgars Rencis** _____ .05.2015.

Recenzents: **M.dat. Jānis Vempers**

Darbs iesniegts 01.06.2015.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: **Darja Solodovņikova** _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2015. prot. Nr. _____

Komisijas sekretārs(-e): _____